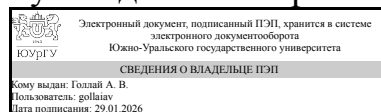


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Руководитель направления



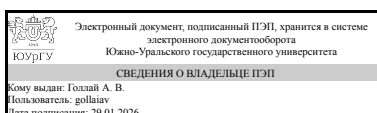
А. В. Голлай

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.21 Разработка Web-приложений
для направления 09.03.01 Информатика и вычислительная техника
уровень Бакалавриат
форма обучения очная
кафедра-разработчик Центр подготовки топ-специалистов в сфере ИТ "Цифровой Урал"

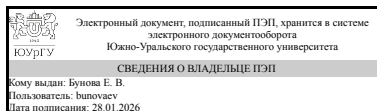
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённым приказом Минобрнауки от 19.09.2017 № 929

Зав.кафедрой разработчика,
д.техн.н., доц.



А. В. Голлай

Разработчик программы,
к.техн.н., доц., доцент



Е. В. Бунова

1. Цели и задачи дисциплины

Целью освоения дисциплины «Разработка Web-приложений» является формирование у студентов целостной системы теоретических знаний и практических навыков, необходимых для проектирования и разработки современных веб-приложений на всех этапах их жизненного цикла. Под жизненным циклом в контексте данной дисциплины понимается последовательная совокупность стадий: выявление и анализ потребностей заказчика и пользователей, формирование и формализация требований к системе, проектирование функций и пользовательских сценариев, разработка прототипов интерфейсов, реализация клиентской (frontend) и серверной (backend) частей, проектирование и использование базы данных, интеграция компонентов клиентской (frontend) и серверной (backend) частей. Для достижения этой цели дисциплина предусматривает изучение следующих ключевых областей и формирование навыков у студентов:

- обучение проектированию веб-приложений, а именно проведению анализа предметной области, выявлять функциональность системы, формализовывать как функциональные, так и нефункциональные требования, а также определять роли пользователей.
- обучение разработке пользовательских сценариев и интерфейсов: студенты освают построение детализированных пользовательских сценариев. На основе этих сценариев студенты научатся проектировать логическую структуру пользовательского интерфейса, определять состав экранов и устанавливать связи между ними.
- обучение прототипированию интерфейсов при использовании современных инструментов проектирования, в частности Figma.
- получение навыков разработки клиентской части веб-приложений (Frontend). Освоение ключевых технологий frontend-разработки: HTML5 и CSS3 для построения и оформления интерфейса, язык JavaScript для реализации динамического поведения, а также фреймворк React как современный подход к разработке frontend-части web-приложений.
- получение навыков разработки серверной части и управление данными (Backend). Обучение принципам построения backend-логики, проектированию модели данных, работе с СУБД PostgreSQL и языком SQL, а также проектированию REST API для эффективного взаимодействия между клиентской и серверной частями приложения.
- получение навыков интеграции и тестирования frontend-части и backend-части.

Краткое содержание дисциплины

Дисциплина «Разработка Web-приложений» нацелена на формирование у студентов комплексных теоретических знаний и практических навыков, необходимых для проектирования и разработки современных веб-приложений на всех этапах их жизненного цикла. Курс охватывает следующие ключевые области:

- 1) Проектирование веб-приложений: от анализа предметной области и формализации функциональных/нефункциональных требований до определения ролей пользователей и системных ограничений.
- 2) Разработка пользовательских сценариев и интерфейсов: построение детализированных сценариев взаимодействия (включая обработку ошибок) и проектирование логической структуры пользовательского интерфейса, состава и связей между экранами.
- 3) Прототипирование интерфейсов: освоение инструментов (в частности, Figma) для создания каркасных прототипов, карт навигации и элементарных дизайн-систем.
- 4) Разработка клиентской части (Frontend): изучение HTML5, CSS3, JavaScript и современного фреймворка React для создания интерактивных и динамичных пользовательских интерфейсов.
- 5)

Разработка серверной части (Backend) и управление данными: освоение принципов построения backend-логики, проектирования моделей данных, работы с СУБД PostgreSQL и SQL, а также создания REST API для взаимодействия с клиентской частью. 6) Интеграция и тестирование: получение навыков по отладке, тестированию и анализу взаимодействия всех компонентов веб-приложения.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	Знает: принципы функционирования web-технологий (HTTP/HTTPS, REST, SOAP, WebSocket); архитектуру клиент-серверных web-приложений и их компонентов; современные стандарты HTML, CSS, JavaScript и принципы адаптивной и кроссбраузерной верстки; основы backend-разработки (обработка запросов, API, базы данных, ORM, авторизация и аутентификация); современные фреймворки и платформы для разработки web-приложений (например, React и др.) Умеет: проектировать архитектуру web-приложений с учётом функциональных и нефункциональных требований; разрабатывать фронтенд- и бэкенд-части web-систем с использованием современных инструментов; интегрировать web-приложения с внешними сервисами и базами данных; использовать системы контроля версий (Git); тестировать и отлаживать web-приложения; применять отечественные и открытые программные средства при реализации проектов Имеет практический опыт: навыками командной разработки web-приложений; практикой применения современных технологий и инструментов web-разработки отечественного и зарубежного производства; умением документировать архитектуру и код web-приложений
ПК-10 Способен выполнять работы по созданию (модификации) и сопровождению информационных ресурсов	Знает: синтаксис выбранного языка web-программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; особенности выбранной среды web-программирования и системы управления базами данных; компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними Умеет: применять выбранные языки web-программирования для написания программного кода ИР; размещать программный код в клиентской и серверной части ИР; оптимизировать программный код ИР с использованием специализированных

	программных средств
--	---------------------

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
1.О.29 Информационные технологии в управлении организационными системами, 1.О.13 Программирование на языках высокого уровня, 1.О.15 Основы теории булевых функций, 1.О.07 Информатика	1.О.24 Основы операционных систем и администрирование Linux

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
1.О.13 Программирование на языках высокого уровня	Знает: основные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня, возможности компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования; возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями Умеет: использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах., применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода; регистрировать изменения исходного текста программного кода в системе управления версиями; сохранять изменения программного кода в соответствии с регламентом управления версиями; выполнять слияние, разделение и сравнение исходных текстов программного кода

	Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux
1.О.29 Информационные технологии в управлении организационными системами	Знает: принципы работы современных информационных технологий; принципы работы программных средств, возможности типовой ИС; инструменты и методы моделирования бизнес-процессов; устройство и функционирование современных ИС; современные подходы и стандарты автоматизации организации Умеет: использовать современные информационные технологий при решении задач профессиональной деятельности; использовать программные средства при решении задач профессиональной деятельности; Имеет практический опыт: использования современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности
1.О.15 Основы теории булевых функций	Знает: теоретические основы и понятийный аппарат теории булевых функций; возможности и область применения аппарата теории булевых функций для понимания принципов работы современных информационных технологий и программных средств; Умеет: применять аппарат теории булевых функций при решении задач профессиональной деятельности; Имеет практический опыт: применения аппарата теории булевых функций при решении задач профессиональной деятельности
1.О.07 Информатика	Знает: принципы работы современных информационных технологий; принципы работы программных средств, базовые понятия информатики и информационных технологий, основные форматы представления информации для автоматизированной обработки; основные принципы работы вычислительных систем и их компонентов; ОПК-2.1. 3-3. Знает основные принципы решения задач с помощью компьютера, понятие алгоритма, основные алгоритмические структуры; состав, назначение функциональных компонентов и программного обеспечения персонального компьютера; методы и приемы алгоритмизации поставленных задач; нотации и программное обеспечение для графического отображения алгоритмов Умеет: использовать современные информационные технологий при решении задач профессиональной деятельности; использовать программные средства при решении задач профессиональной деятельности, работать с

	информацией, представленной в различных формах; разрабатывать алгоритмы для решения типовых задач; работать с персональным компьютером, применять современное ПО, в т.ч., отечественного производства, для решения практических задач, использовать методы и приемы формализации и алгоритмизации поставленных задач Имеет практический опыт: навыками использования современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности, разработки алгоритмов для решения типовых задач
--	---

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 4 з.е., 144 ч., 104,75 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах	
		Номер семестра	
		3	4
Общая трудоёмкость дисциплины	144	72	72
<i>Аудиторные занятия:</i>	96	48	48
Лекции (Л)	32	16	16
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	64	32	32
Лабораторные работы (ЛР)	0	0	0
<i>Самостоятельная работа (СРС)</i>	39,25	19,75	19,5
Изучение литературы	39,25	19,75	19,5
Консультации и промежуточная аттестация	8,75	4,25	4,5
Вид контроля (зачет, диф.зачет, экзамен)	-	зачет	диф.зачет

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Основные принципы разработки веб-приложений и управления интернет-проектами. Жизненный цикл Web-систем. Систематизация требований к разрабатываемому веб-приложению. Архитектура «клиент–сервер» и модель «клиент–API–сервер»	8	8	0	0
2	Проектирование веб-интерфейсов. Прототипирование в Figma	8	2	6	0
3	Frontend. Понятие фронтенд-разработки и её место в структуре веб-приложения. Основы HTML5. CSS3.	10	2	8	0
4	Frontend. Язык JavaScript в современной веб-разработке и его места в структуре клиентской части веб-приложения	14	4	10	0

5	Frontend. Библиотека React в современной фронтенд-разработке и её роли в создании пользовательских интерфейсов веб-приложений. Введение в React	10	2	8	0
6	Backend: Серверная часть веб-приложения и роли backend-разработки в архитектуре web-систем. Основы PHP	16	4	12	0
7	Роль базы данных в структуре веб-приложения и её значения для хранения и обработки информации.	10	4	6	0
8	Сборка пользовательского интерфейса, серверной части и базы данных в единую систему. Тестирование веб-приложения. Система контроля версий. Понятия репозитория, коммита, ветвления, слияния. Организация репозитория учебного проекта, структура каталогов, принципы ведения истории изменений и работы в мини-команде.	20	6	14	0

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Понятие веб-приложения и его место в современных информационных системах. Назначение веб-приложений, области применения, примеры использования. Понятие жизненного цикла web-системы. Общая характеристика этапов. Этап инициирования проекта. Формирование идеи и целей веб-системы. Этап анализа и формализации требований. Определение функциональных и нефункциональных требований. Этап проектирования web-системы. Разработка архитектуры и интерфейсов. Этап разработки веб-приложения. Программная реализация системы. Этап тестирования web-системы. Проверка корректности и надёжности работы. Этап внедрения веб-приложения. Размещение системы в рабочей среде. Этап эксплуатации и сопровождения web-системы. Поддержка, обновление и развитие.	4
2	1	Систематизация требований к разрабатываемому веб-приложению. Функциональные и нефункциональные требования. Процесс формирования функций на основе описания предметной области. Концепция пользовательских ролей и прав доступа; разделение функциональности по ролям (пользователь, администратор, модератор). Переход от перечня функций к пользовательским сценариям, в которых фиксируются последовательности шагов, альтернативные ветви и ошибки. Сформированный набор требований и сценариев как основа для дальнейшего проектирования интерфейса, архитектуры и данных.	2
3	1	Архитектура «клиент–сервер» и модель «клиент–API–сервер». Протокол HTTP и формат JSON. Базовая архитектура веб-приложений: классическая модель «клиент–сервер» и распространённая модель «клиент–API–сервер», в которой frontend-приложение взаимодействует с backend через программный интерфейс (REST API). Протокол HTTP: структура запросов и ответов, методы (GET, POST, PUT, DELETE и др.), коды состояния (2xx, 3xx, 4xx, 5xx), заголовки и тело сообщений. Формат JSON - стандарт для представления данных в веб-приложениях, его преимущества по сравнению с традиционными форматами. Пользовательские действия в интерфейсе и соответствующими им запросы к API, REST-подход при проектировании backend-части.	2
4	2	Прототипирование пользовательского интерфейса и пользовательских сценариев. Информационная структура веб-интерфейса. Навигация, меню, экраны, переходы между разделами. Среда Figma как инструмент прототипирования. Назначение, возможности и область применения. Связь прототипирования с дальнейшей вёрсткой и программной реализацией интерфейса.	2

5	3	Понятие фронтенд-разработки и её место в структуре веб-приложения. Роль HTML и CSS в создании пользовательского интерфейса. Язык разметки HTML5. Структура HTML-документа. Основные теги и их назначение. Семантические элементы HTML5. Их роль в структуре страницы, доступности и поисковой оптимизации. Понятие каскадных таблиц стилей CSS3. Назначение CSS в оформлении веб-страниц. Основные способы подключения CSS. Селекторы, классы и идентификаторы. Блочная модель CSS. Основы адаптивной вёрстки в CSS3. Связь HTML5 и CSS3 при создании пользовательского интерфейса.	2
6	4	Язык JavaScript в современной веб-разработке и его места в структуре клиентской части веб-приложения. Рассмотрение базовых понятий языка JavaScript: переменные, типы данных, операторы, выражения. Изучение условных операторов, циклов и логических конструкций. Рассмотрение функций как основы построения программной логики. Понятие области видимости переменных. Основы работы с массивами и объектами. Понятие DOM (Document Object Model) как объектной модели HTML-документа. Структура DOM-дерева. Методы доступа к элементам страницы. Событийная модель JavaScript. Основные типы событий. Асинхронность в JavaScript. Основы работы с HTTP-запросами. Использование функции fetch для получения данных с сервера. Понятие формата JSON и его роль в обмене данными между клиентом и сервером. Общие принципы взаимодействия JavaScript с серверной частью веб-приложения.	4
7	5	Библиотека React в современной фронтенд-разработке и её роли в создании пользовательских интерфейсов веб-приложений. React как библиотека для построения компонентных интерфейсов. Основные принципы работы React: декларативный подход, виртуальный DOM, повторное использование компонентов. Понятие компонента как основной единицы построения интерфейса. Классовые и функциональные компоненты, их назначение и особенности. Понятие JSX как расширения синтаксиса JavaScript для описания интерфейса. Передача данных в компоненты с использованием props. Понятие состояния компонента и общий принцип его изменения. Обзор хука useState. Рендеринг компонентов и обновление интерфейса при изменении данных. Связь React с ранее изученными технологиями HTML, CSS и JavaScript. Области применения React при разработке одностраничных веб-приложений. Механизмы управления состоянием в React с использованием функциональных компонентов и хуков (useState, useEffect). Взаимодействие компонентов с REST API: выполнение запросов к серверу, обработка ответов и ошибок, отображение состояний загрузки. Организация потока данных в приложении: от действий пользователя к запросам к API и обратно к обновлению интерфейса.	2
8	6	Серверная часть веб-приложения и роли backend-разработки в архитектуре web-систем. REST API с позиции архитектуры backend. Анализируется модель «клиент–API–сервер» как целевая для дисциплины: frontend-приложение на React взаимодействует с backend, реализованным на любой современной платформе (PHP, Node.js, .NET, Python и др.), через стандартизированный интерфейс. Рассматриваются принципы проектирования ресурсов, маршрутов, форматов запросов и ответов. Обсуждается версионирование API, стабильность контрактов, документация (например, OpenAPI/Swagger) как элемент промышленной разработки. Подчёркивается технологическая нейтральность концепции: конкретный язык и фреймворк backend не фиксируются, акцент делается на архитектурных принципах. Рассмотрение языка программирования PHP как одного из основных инструментов серверной разработки. Области применения PHP, особенности интерпретируемого исполнения. Основы синтаксиса PHP: подключение PHP-скриптов, вывод данных, переменные и типы данных. Операторы и выражения в PHP. Условные конструкции if,	4

		switch. Циклы for, while, foreach. Понятие массивов в PHP, одномерные и многомерные массивы. Ассоциативные массивы. Функции в PHP, пользовательские и встроенные функции. Организация структуры PHP-проекта	
9	7	Роль базы данных в структуре веб-приложения и её значения для хранения и обработки информации. Понятие базы данных и системы управления базами данных. Реляционная модель данных. Основные понятия: таблица, запись, поле, первичный ключ. Назначение и особенности СУБД PostgreSQL. Понятие SQL как языка управления данными. Классификация SQL-запросов. Основные запросы выборки, добавления, изменения и удаления данных. Понятие связей между таблицами. Первичные и внешние ключи. Обеспечение целостности данных. Использование соединений таблиц (JOIN). Основы проектирования структуры базы данных для веб-приложения. Понятие индексов. Общие принципы оптимизации работы с базами данных. Взаимодействие серверной части веб-приложения с базой данных. Формирование данных для передачи на клиент в формате JSON.	4
10	8	Интеграция компонентов веб-приложения и её значения для создания работоспособной web-системы. Взаимосвязь фронтенда, серверной части и базы данных в единой архитектуре веб-приложения. Общие принципы взаимодействия клиентской и серверной частей через API. Понятие точек доступа (endpoint). Передача данных между фронтендом и сервером в формате JSON. Роль базы данных в обеспечении целостности и сохранности информации. Последовательность сборки компонентов веб-приложения в единую систему. Настройка среды выполнения и серверной конфигурации для размещения веб-приложения. Понятие тестирования веб-приложений и его место в жизненном цикле web-систем. Система контроля версий. Понятия репозитория, коммита, ветвления, слияния. Организация репозитория учебного проекта, структура каталогов, принципы ведения истории изменений и работы в мини-команде.	6

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	2	Проектирование веб-приложений: определение целей и задач, целевой аудитории, функциональных требований и пользовательских сценариев. Моделирование поведения пользователя. Основы UI/UX и принципы проектирования: различие понятий UI и UX, практическое проектирование удобных и понятных интерфейсов. Отработка логики расположения элементов, визуальной иерархии и единого стилевого оформления. Разработка информационной структуры: формирование структуры экранов, меню и навигационных переходов между разделами приложения. Использование Figma для создания фреймов и экранов. Размещение текста, кнопок, форм, иконок, изображений. Использование сеток и направляющих. Прототипирование пользовательских сценариев. Настройка интерактивных переходов между экранами. Тестирование прототипа и проверка логики интерфейса.	6
2	3	Практическое знакомство с понятием фронтенд-разработки и её местом в структуре веб-приложения, определение ролей HTML и CSS в формировании пользовательского интерфейса. Изучение структуры HTML5-документа, обязательных служебных тегов, правил вложенности элементов. Практическое использование базовых тегов для разметки текста, изображений, ссылок, списков и таблиц. Освоение семантических элементов HTML5 для построения логичной структуры страницы с учётом требований	4

		доступности и поисковой оптимизации. Знакомство с понятием каскадных таблиц стилей CSS3 и их назначением в оформлении веб-страниц. Практическая работа со способами подключения CSS: встроенные, внутренние и внешние стили. Использование селекторов различных типов, классов и идентификаторов для управления оформлением элементов.	
3	3	Изучение блочной модели CSS, настройка размеров элементов, внутренних и внешних отступов, границ и фона. Практическое применение свойств позиционирования элементов. Практическая отработка взаимодействия HTML5 и CSS3 при создании целостного пользовательского интерфейса веб-приложения.	4
4	4	Основы синтаксиса JavaScript: объявления переменных с использованием var, let и const, типы данных, операторы, выражения и преобразование типов. Изучение условных операторов if, switch, логических операций и ветвлений. Изучение циклов for, while, do...while и их применение. Рассмотрение функций, стрелочных функций и области видимости переменных. Основы работы с массивами и методами массивов. Основы работы с объектами JavaScript.	6
5	4	Понятие DOM (Document Object Model) как объектной модели HTML-документа. Структура DOM-дерева. Методы доступа к элементам страницы. Событийная модель JavaScript и основные типы событий. Понятие асинхронности в JavaScript. Использование промисов. Основы работы с HTTP-запросами. Использование функции fetch для получения данных с сервера. Понятие формата JSON, сериализация и десериализация данных. Общие принципы взаимодействия JavaScript с серверной частью веб-приложения.	4
6	5	Инициализация проекта на React и подготовка структуры репозитория (Git). Создание нового проекта на React, инициализация Git-репозитория. Определяется структура каталогов для компонентов, страниц, стилей и вспомогательных модулей. Настраивается базовый роутинг для нескольких тестовых страниц. Фиксирование этапов развития проекта в системе контроля версий. Подключение React к проекту, запуск и структура минимального React-приложения. Реализация в React основных структурных компонентов интерфейса для основных экранов и общих элементов интерфейса: шапки, меню, подвала, базовых контейнеров. Вывод статического содержимого в интерфейс с использованием JSX. Практическая передача данных в компонент через props. Создание нескольких компонентов и организация их иерархии. Реализация простых интерактивных элементов интерфейса на React. Обновление интерфейса при изменении состояния. Обработка пользовательских событий в React-компонентах. Использование ранее изученных CSS-стилей в React-приложении.	6
7	5	REST API: получение списков и подробной информации по сущностям. Реализуются запросы с помощью fetch (или другой библиотеки по согласованию), результат обрабатывается и отображается в компонентах списков и карточек. Добавляются индикаторы загрузки и простейшая обработка ошибок. разрабатывают формы создания, редактирования и удаления сущностей и связывают их с соответствующими вызовами API (POST, PUT/PATCH, DELETE). Обрабатываются успешные и ошибочные ответы. В интерфейсе выполняется обновление списков и состояний после изменения данных. Результатом становится полноценная реализация цикла CRUD на стороне клиента. Проектирование и уточнение структуры REST API для учебного проекта. На основе реализованного интерфейса и сценариев уточнение структуры REST API: перечень ресурсов, маршрутов, форматов запросов и ответов. Фиксируются требования к стабильности контракта и обратной совместимости.	2

8	6	Практическая установка и настройка локального сервера для выполнения PHP-скриптов. Создание и запуск первых PHP-скриптов. Практическое использование операторов вывода данных. Работа с переменными различных типов. Реализация условий и циклов в прикладных задачах. Практическая работа с массивами и ассоциативными массивами. Создание пользовательских функций и их использование в программе. Подключение файлов и разбиение программы на модули. Обработка данных, передаваемых из HTML-форм методом GET и POST. Практическая работа с суперглобальными массивами.	6
9	6	Реализация простых серверных сценариев обработки пользовательских запросов. Формирование и отправка данных в формате JSON. Приём данных от клиентской части веб-приложения. Связь PHP-логики с ранее созданным фронтенд-интерфейсом. Отладка PHP-кода и обработка ошибок.	6
10	7	Установка и настройка СУБД PostgreSQL и инструмента администрирования pgAdmin. Создание базы данных и таблиц. Определение полей, типов данных, первичных ключей и ограничений. Практическая работа с SQL-запросами SELECT, INSERT, UPDATE и DELETE. Наполнение таблиц тестовыми данными. Реализация связей между таблицами с использованием внешних ключей. Практическое применение JOIN для получения связанных данных. Получение данных из базы и формирование ответов в формате JSON. Передача данных на фронтенд и их отображение в пользовательском интерфейсе. Практическая отладка запросов и анализ ошибок при работе с базой данных. Обеспечение базовой безопасности при работе с данными.	6
11	8	Практическая интеграция ранее разработанного пользовательского интерфейса с серверной частью веб-приложения. Настройка обмена данными между фронтендом и сервером с использованием HTTP-запросов. Практическое подключение базы данных к серверной части и проверка корректности получения и сохранения данных. Реализация полного пользовательского сценария: ввод данных через интерфейс, передача на сервер, обработка и сохранение в базе данных, возврат результата пользователю. Проверка корректности работы API. Практическая отладка взаимодействия между всеми компонентами системы.	4
12	8	Проведение функционального тестирования веб-приложения. Проверка основных пользовательских сценариев. Тестирование форм ввода данных и обработки ошибок. Проверка корректности работы системы авторизации (при наличии). Использование Git при разработке. Ветки, коммиты, работа над учебным проектом.	4
13	8	Проектная практика: разработка пользовательского интерфейса с интеграцией с серверной частью веб-приложения. Использование Git при разработке. Ветки, коммиты, работа над учебным проектом. Выполняются операции создания веток для новых функциональностей, В контексте учебного проекта формируется история изменений, связывающая этапы разработки с зафиксированными коммитами.	6

5.3. Лабораторные работы

Не предусмотрены

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов

Изучение литературы	Позевалкин, В. В. Разработка веб-приложений на основе клиентских каркасов и библиотек : учебное пособие / В. В. Позевалкин, Н. Ф. Панова. — Оренбург : ОГУ, 2025. — 191 с. — ISBN 978-5-7410-3380-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/502621 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей.	4	19,5
Изучение литературы	Калиберда, Е. А. Разработка web-приложений : учебное пособие / Е. А. Калиберда, К. В. Кравченко. — Омск : ОмГТУ, 2023. — 100 с. — ISBN 978-5-8149-3679-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/421766 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользовате	3	19,75

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-местр	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учитывается в ПА
1	4	Промежуточная аттестация	4 семестр Итоговая проектная работа	-	5	5 баллов — пользовательский интерфейс полностью реализован и корректно работает во всех основных браузерах, реализованы все экраны, формы ввода, передача и отображение данных, выполнена базовая клиентская валидация и логичная навигация. Серверная часть на PHP обеспечивает приём, обработку и возврат данных в формате JSON, выполнена минимальная валидация и базовая защита от SQL-инъекций. Создана и подключена база данных PostgreSQL через PDO, данные корректно сохраняются и извлекаются. Реализован полный	дифференцированный зачет

					пользовательский сценарий (ввод → сервер → БД → возврат пользователю). Проект уверенно продемонстрирован, архитектура и распределение ролей объяснены, даны ответы на вопросы. 4 балла — основные компоненты системы реализованы и работают, однако имеются незначительные недочёты в интерфейсе, серверной логике или структуре базы данных, не влияющие критически на работоспособность. Полный пользовательский сценарий в целом выполняется. Защита проекта проведена с отдельными неточностями в пояснениях. 3 балла — реализована только базовая функциональность: интерфейс упрощён, серверная часть и база данных работают частично, пользовательский сценарий выполняется с ошибками или ограниченно. Интеграция компонентов неполная. Защита проекта проведена с существенными затруднениями. 2 балла — проект неработоспособен либо реализован фрагментарно: отсутствует корректная связь между интерфейсом, сервером и базой данных, пользовательский сценарий не выполняется, студент не может пояснить принцип работы системы. 1 балла — проект неработоспособен 0 балла — проект неработоспособен		
2	3	Текущий контроль	Контрольная работа 1	1	5	Сводные критерии оценки контрольного задания 1 (0–5 баллов) 5 баллов — разработана полная и логически выстроенная структура интерфейса, определены все основные экраны и их	зачет

						назначение, пользовательские сценарии проработаны полностью и корректно, навигация удобная и непротиворечивая. В Figma создан полноценный интерактивный прототип со всеми ключевыми элементами и переходами. Прототип полностью соответствует заданной предметной области и может быть использован для дальнейшей разработки. 4 балла — структура интерфейса и сценарии в целом разработаны корректно, прототип в Figma создан, но отдельные экраны, элементы или переходы проработаны упрощённо либо с незначительными логическими неточностями. 3 балла — представлена только базовая структура интерфейса, пользовательские сценарии описаны частично, навигация содержит ошибки или не отражает полностью логику работы системы, прототип в Figma выполнен в минимальном объёме. 2 балла — структура интерфейса фрагментарна или нелогична, пользовательские сценарии отсутствуют либо не соответствуют заданию, прототип в Figma не создан либо не отражает принципов работы веб-приложения. 1 балла — прототип в Figma не создан 0 балла — прототип в Figma не создан	
3	3	Текущий контроль	Контрольная работа 2	1	5	5 баллов — HTML-разметка полностью соответствует прототипу, корректно использованы семантические теги, формы и навигационные элементы. Структура документа логична и удобочитаема. CSS-оформление выполнено аккуратно, применены шрифты, цвета, отступы и блочная модель, интерфейс визуально согласован.	зачет

					элементы не «едут» и не накладываются. Реализована базовая адаптивность (интерфейс корректно отображается на разных разрешениях экрана). В целом интерфейс можно считать готовым к дальнейшей разработке. 4 балла — Основная HTML-структура реализована корректно, семантические теги в целом используются правильно, прототип в основном соблюден, но возможны отдельные отклонения (упрощенные элементы, не все мелкие детали учтены). CSS-оформление в целом аккуратное, однако присутствуют незначительные визуальные или адаптивные недочёты (например, неидеальное поведение на узких экранах, небольшие проблемы с отступами или выравниванием). 3 балла — HTML-разметка выполнена частично: структура страниц упрощена, семантика используется ограниченно или непоследовательно, соответствие прототипу неполное. CSS-оформление реализовано на базовом уровне: интерфейс выглядит рабочим, но содержит заметные визуальные неточности (неровные отступы, «прыгающие» элементы, отсутствие или формальная адаптивность). Для дальнейшей разработки макет требует доработки. 2 балла — HTML-разметка фрагментарна или структурно неверна, семантика практически не соблюдена, соответствие прототипу слабое или отсутствует. CSS-оформление выполнено частично: интерфейс выглядит «сырой», элементы накладываются, отступы и	
--	--	--	--	--	---	--

						размеры не выстроены, адаптивность отсутствует. Использовать результат как основу для реального веб-приложения нельзя без существенной переработки. 1 балл — HTML-разметка неверна, CSS-оформление отсутствует 0 баллов — HTML-разметка неверна, CSS-оформление отсутствует	
4	3	Текущий контроль	Контрольная работа 3	1	5	5 баллов — реализована полноценная обработка пользовательских событий (кнопки, формы, элементы выбора), корректная клиентская валидация данных, выполнена динамическая работа с DOM (добавление, изменение, удаление элементов). Скрипты работают устойчиво, без критических ошибок, интерфейс корректно реагирует на все действия пользователя. Ошибочные действия обрабатываются корректно с выводом понятных сообщений. 4 балла — основные события и валидация реализованы корректно, динамическая работа с DOM присутствует, однако имеются незначительные логические ошибки или упрощения, не влияющие критически на работоспособность интерфейса. Возможны отдельные недочёты в обработке нестандартных ситуаций. 3 балла — реализована только базовая обработка событий и частичная валидация данных. Динамическое изменение DOM выполнено в минимальном объёме. Интерфейс реагирует на действия пользователя, но содержит заметные логические ошибки или нестабильные элементы. 2 балла — JavaScript-логика реализована фрагментарно	зачет

						либо работает некорректно. Обработка пользовательских событий неполная или отсутствует, валидация данных не выполнена, динамическая работа с DOM минимальна или отсутствует. Работа интерфейса нестабильна. 1 балл — обработка пользовательских событий отсутствует 0 баллов — обработка пользовательских событий отсутствует	
5	3	Текущий контроль	3 семестр Итоговая контрольная работа (командная работа)	1	5	5 баллов — реализован полностью работоспособный интерактивный сервис, корректно функционируют все пользовательские сценарии, интерфейс удобен, логика устойчива, вклад участников сбалансирован. 4 балла — проект в целом работоспособен, имеются незначительные логические или визуальные недочёты. 3 балла — реализована только базовая часть функциональности, сценарий отработан частично. 2 балла — проект фрагментарен, пользовательский сценарий реализован некорректно либо не реализован. 1 балл — интерактивный сервис не реализован 0 баллов — интерактивный сервис не реализован	зачет
6	4	Текущий контроль	Контрольная работа 6	1	5	5 баллов. Работа выполнена полностью и без методических, логических и технических ошибок. Серверная среда настроена корректно. Все этапы задания реализованы в полном объёме. Переменные, условия, циклы, массивы и пользовательские функции используются правильно. Программа имеет модульную структуру. Вычисления корректны. Код структурирован, читаем, сопровождается	дифференцированный зачет

						комментариями. Отчёт полный, логически связанный и грамотно оформлен. 4 балла. Все основные этапы выполнены, однако допущены отдельные несущественные недочёты: упрощённая модульная структура, избыточные конструкции, частичное дублирование кода. Программа функционирует корректно, отчёт представлен в полном объёме, но содержит поверхностные описания. 3 балла. Реализованы только базовые элементы задания: переменные, условия, циклы. Работа с массивами или функциями выполнена формально либо частично. Модульность отсутствует. Программа выполняется, но имеет упрощённую логику. Отчёт содержит существенные пропуски. 2 балла. Реализованы отдельные фрагменты программы без цельной логики. Серверная среда настроена с ошибками. Программа функционирует нестабильно. Отчёт неполный. 1 балл. Представлены разрозненные элементы кода без логической связи. Работа не демонстрирует освоение базовых принципов PHP-разработки. Отчёт формальный. 0 баллов. Работа не выполнена или не представлена. Программа не запускается.	
7	4	Текущий контроль	Контрольная работа 7	1	5	5 баллов. Работа выполнена полностью в соответствии с заданием и в полном объёме. Разработана корректная, логически структурированная HTML-форма, включающая все требуемые элементы управления, форма корректно отправляет данные на сервер как методом GET, так и методом POST. Серверная	дифференцированный зачет

					часть на PHP правильно принимает данные через суперглобальные массивы \$_GET и \$_POST, различает способы передачи, корректно обрабатывает отсутствие и заполненность параметров. Грамотно использован массив \$_SERVER для получения служебных параметров HTTP-запроса с их осмысленным анализом в отчёте. В полном объёме реализована валидация данных: очистка пользовательского ввода, проверка типов, диапазонов и длины строк, защита от внедрения HTML-кода и сценариев. Прикладная логика реализована корректно, результаты обработки отображаются в структурированной, понятной и безопасной форме. Код программы логически структурирован, читаем, аккуратно оформлен, имена переменных и файлов осмысленны. Отчёт полный, содержит описание всех этапов, пояснения к коду, скриншоты и выводы. 4 балла. Работа в основном выполнена правильно и соответствует требованиям задания. HTML-форма корректно реализована и отправляет данные на сервер, приём данных через GET и POST работает, однако различие между методами может быть раскрыто не полностью. Суперглобальные массивы используются корректно, но демонстрация или пояснение их содержания выполнены поверхностно. Валидация и защита реализованы частично: присутствует очистка ввода и защита от HTML-вставок, но не все поля проверены на тип, диапазон или длину. Прикладная логика обработки данных реализована, вывод	
--	--	--	--	--	---	--

					результатов корректен, но может быть упрощён по оформлению. Код в целом читаем, однако имеются отдельные стилистические или структурные недочёты. Отчёт представлен в полном объёме, но отдельные этапы описаны без достаточной глубины. 3 балла. Работа выполнена частично. HTML-форма в целом присутствует, но реализована формально, без продуманной структуры или без полного набора элементов. Приём данных реализован только для одного метода (чаще всего POST), либо обработка второго метода носит формальный характер. Использование суперглобальных массивов ограничивается базовым извлечением значений без анализа структуры и без осмысленного использования \$_SERVER. Валидация данных выполнена минимально либо отсутствует для части полей, защита от HTML-вставок реализована частично или формально. Прикладная обработка данных упрощена, вывод результатов возможен, но не всегда аккуратен и информативен. Код работает, но содержит признаки фрагментарности и слабой структурированности. Отчёт неполный, отдельные этапы выполнения описаны поверхностно. 2 балла. Работа выполнена на низком уровне. HTML-форма содержит ошибки либо не соответствует заданию по составу элементов. Приём данных через PHP реализован нестабильно, возможны частые ошибки при отсутствии параметров, методы GET и POST не	
--	--	--	--	--	---	--

						различаются. Суперглобальные массивы используются фрагментарно, без понимания их назначения. Полноценная валидация и защита данных отсутствуют либо носят номинальный характер. Прикладная логика обработки данных выражена слабо или реализована с ошибками. Вывод результатов может быть некорректным, неструктурированным, содержать технические сообщения. Код плохо структурирован, содержит логические и синтаксические ошибки. Отчёт неполный, не отражает ключевые этапы выполнения работы. 1 балл. Представлены лишь отдельные фрагменты формы и/или серверного скрипта без целостной логики работы. Передача данных от формы к серверу реализована формально либо работает нестабильно. Суперглобальные массивы используются случайно, без осознания их назначения. Валидация и защита отсутствуют. Прикладная задача фактически не реализована. Код несистематизированный, содержит множество ошибок. Отчёт носит формальный характер либо практически отсутствует. 0 баллов. Контрольная работа не представлена либо представлена в виде неработоспособного набора файлов. HTML-форма и серверная часть отсутствуют или не запускаются. Отчёт отсутствует. Попыток осмысленной реализации задания не выявлено.	
8	4	Текущий контроль	Контрольная работа 8	1	5	5 баллов. Контрольная работа выполнена полностью, в строгом соответствии с	дифференцированный зачет

					заданием и в полном объёме. Реализованы не менее двух–трёх серверных эндпоинтов, каждый из которых корректно принимает входные данные, выполняет прикладную обработку и возвращает структурированный ответ в формате JSON. Чётко соблюдена архитектура «frontend – API – backend». Формат JSON-ответов унифицирован, корректно установлены заголовки Content-Type, отсутствуют посторонние сообщения PHP в теле ответа. Интеграция с фронтендом реализована полностью: клиентская часть корректно формирует запросы, принимает JSON-ответы, обрабатывает данные и отображает их пользователю. В коде серверных сценариев реализована полноценная проверка входных данных, предусмотрена обработка исключительных ситуаций, формируются осмысленные сообщения об ошибках. Отладка и логика работы устойчивы при некорректных запросах. Отчёт полный, содержит описание API, примеры запросов и ответов, листинги кода, скриншоты успешных и ошибочных сценариев, выводы. Работа демонстрирует системное понимание принципов построения серверного API и обмена данными с клиентской частью. 4 балла. Контрольная работа выполнена в целом правильно, реализованы два серверных эндпоинта, формирование и возврат данных в формате JSON осуществляется корректно, однако отдельные элементы архитектуры или обработки ошибок реализованы упрощённо. Интеграция с фронтендом выполнена,	
--	--	--	--	--	--	--

					запросы отправляются и ответы корректно принимаются, но обработка ошибок на клиентской стороне может быть частичной. Проверка входных данных на сервере присутствует, однако охватывает не все возможные некорректные ситуации либо сообщения об ошибках недостаточно информативны. JSON-обмен корректен, но структура ответов может быть не полностью унифицированной. Отчёт представлен в полном объёме, но отдельные разделы раскрыты поверхностно, без глубокого анализа. В целом работа демонстрирует уверенное понимание темы, но содержит отдельные методические и технические недочёты. 3 балла. Контрольная работа выполнена частично. Реализован один рабочий эндпоинт либо два, но с ограниченной функциональностью. Формирование JSON-ответов осуществляется, однако структура ответов нестабильна, возможны технические сообщения в выводе. Интеграция с фронтендом реализована в упрощённом виде: запросы отправляются, но обработка полученных данных или ошибок ограничена. Проверка входных данных на сервере минимальна или формальна, сообщения об ошибках не всегда корректны. Серверные сценарии работают, но устойчивость при некорректных запросах не гарантирована. Отчёт представлен не полностью, содержит лишь основные сведения без детального описания структуры API и обмена данными. Работа демонстрирует базовое	
--	--	--	--	--	--	--

					понимание принципов JSON-обмена, но без уверенного владения методами. 2 балла. Контрольная работа выполнена на низком уровне. Серверная часть реализована фрагментарно, количество эндпоинтов недостаточно либо они работают нестабильно. Формирование JSON-ответов выполняется с ошибками, возможен вывод HTML или служебных сообщений вместо корректного JSON. Интеграция с фронтом формальная или нестабильная: данные передаются нерегулярно, пользовательский интерфейс не всегда получает корректный результат. Проверка входных данных на сервере практически отсутствует, обработка ошибок не организована либо носит случайный характер. Отчёт неполный, не отражает в достаточной мере структуру API и логику взаимодействия компонентов. Работа не демонстрирует целостного понимания принципов построения серверных сценариев. 1 балл. Представлены лишь отдельные фрагменты серверного и клиентского кода без целостной архитектуры API. Реализация эндпоинтов формальна либо работает с грубыми ошибками. JSON-обмен реализован частично или некорректно. Интеграция с фронтом либо отсутствует, либо не функционирует. Проверка входных данных и обработка ошибок не реализованы. Отчёт носит формальный характер либо практически отсутствует. Работа не позволяет сделать вывод о сформированности навыков серверного взаимодействия с клиентской частью.	
--	--	--	--	--	---	--

						0 баллов. Контрольная работа не представлена либо представлена в виде неработоспособного набора файлов. Серверные сценарии отсутствуют или не запускаются. JSON-обмен не реализован. Связь с фронтендом отсутствует. Отчёт отсутствует. Попыток осмысленной реализации задания не выявлено.	
9	4	Текущий контроль	контрольное задание 9	1	5	5 (отлично) – Полностью реализована база данных, верно спроектированы таблицы и связи, выполнены все типы SQL-запросов, данные корректно передаются в формате JSON, обеспечена защита от SQL-инъекций, отчёт полный и аргументированный. 4 (хорошо) – База данных реализована корректно, но имеются частичные недочёты в связях, запросах или защите, JSON-формирование выполнено без системных ошибок, отчёт достаточный. 3 (удовлетворительно) – Реализована структура БД и базовые запросы SELECT/INSERT, сложные JOIN и безопасность раскрыты слабо, отчёт неполный. 2 (неудовлетворительно) – Структура БД фрагментарна, запросы выполняются нестабильно, отсутствует защита, отчёт формальный. 1 (низкий уровень) – Представлены отдельные SQL-фрагменты без целостной модели данных. 0 – Работа не выполнена.	дифференцированный зачет
11	4	Текущий контроль	Контрольная работа 11	1	5	5 баллов (отлично) Интеграция клиентской части, серверного API и базы данных реализована полностью и устойчиво. Реализован один или несколько полноценных сквозных пользовательских сценариев, корректно функционирующих при различных входных данных.	дифференцированный зачет

					Обмен данными между frontend и backend осуществляется строго в формате JSON с корректно установленными заголовками. Все этапы передачи, обработки и отображения данных реализованы без логических и технических ошибок. Предусмотрена полноценная обработка ошибок и исключительных ситуаций. Все компоненты системы согласованы по форматам и структурам данных. Отчёт полный, структурированный, содержит все необходимые схемы, листинги, примеры запросов и ответов, скриншоты и аналитические выводы. Работа демонстрирует системное понимание принципов интеграции веб-приложений. 4 балла (хорошо) Интеграция frontend, backend и базы данных в целом реализована корректно. Основной пользовательский сценарий работает устойчиво, обмен JSON-данными осуществляется правильно. Однако при нестандартных ситуациях (неполные данные, ошибки соединения, граничные значения) могут возникать отдельные сбои или недостаточно информативные сообщения об ошибках. Структура взаимодействия компонентов в целом выстроена правильно, но устойчивость системы ограничена. Отчёт содержит все основные элементы, однако отдельные этапы раскрыты без достаточной детализации. 3 балла (удовлетворительно) Реализован один сквозной пользовательский сценарий, однако интеграция носит частичный характер. Обмен данными между фронтендом и сервером осуществляется,	
--	--	--	--	--	---	--

					но структура JSON-ответов может быть нестабильной. Обработка ошибок реализована формально либо частично. Возможны случаи некорректной обработки данных. Отчёт представлен не в полном объёме, отдельные элементы взаимодействия компонентов описаны поверхностно. Работа демонстрирует базовое понимание принципов интеграции, но без устойчивой реализации. 2 балла (неудовлетворительно) Связь между frontend и backend реализована нестабильно. Отдельные запросы выполняются, однако целостный пользовательский сценарий не обеспечен. Форматы данных согласованы частично, JSON-обмен нарушается при возникновении ошибок. Работа системы зависит от внешних условий, возможны частые сбои. Отчёт неполный, схемы взаимодействия отсутствуют или представлены формально. Работа не демонстрирует устойчивых навыков интеграции. 1 балл (крайне низкий уровень) Интеграция между клиентской и серверной частью носит формальный характер. Представлены лишь отдельные элементы взаимодействия без целостной архитектуры. Пользовательский сценарий не реализован в завершённом виде. Обработка данных и ошибок не организована. Отчёт формальный либо практически отсутствует. 0 баллов (работа не выполнена) Контрольная работа не представлена либо представлена в виде	
--	--	--	--	--	--	--

						неработоспособного набора файлов. Связь между frontend, backend и базой данных отсутствует. Отчёт отсутствует. Попыток осмысленной реализации интеграции не выявлено.	
12	4	Текущий контроль	Контрольная работа 12	1	5	5 баллов (отлично). Контрольная работа выполнена полностью и на высоком профессиональном уровне. Подготовлен полный и корректный набор тест-кейсов, охватывающий все ключевые пользовательские сценарии и граничные случаи. Функциональное тестирование проведено системно, все тесты выполнены и задокументированы. Выявленные дефекты грамотно классифицированы, причины их возникновения проанализированы, выполнено полное устранение ошибок с последующим повторным тестированием. Использование системы Git организовано корректно: история коммитов логична, сообщения осмысленны, применялись ветки и слияния. Отчёт полный, структурированный, содержит тест-кейсы, журнал ошибок, историю коммитов, скриншоты и аналитические выводы. Работа демонстрирует сформированные профессиональные навыки контроля качества программного продукта. 4 балла (хорошо). Работа выполнена в целом корректно. Подготовлен достаточный набор тест-кейсов, охватывающий основные функциональные сценарии. Тестирование проведено, выявленные ошибки задокументированы и частично устранены. Использование Git	дифференцированный зачет

					организовано правильно, но история коммитов может быть недостаточно детализированной либо с редким применением ветвлений. Отчёт содержит все основные разделы, однако отдельные элементы анализа представлены поверхностно. Работа демонстрирует уверенное, но не полностью устойчивое владение методами функционального тестирования и контроля версий. 3 балла (удовлетворительно). Подготовлен ограниченный набор тест-кейсов, охватывающий только базовые сценарии. Тестирование проведено частично, не все ошибки выявлены и задокументированы. Исправления выполнены выборочно. Использование Git носит формальный характер: коммиты редкие, ветвления отсутствуют либо не используются по назначению. Отчёт представлен не в полном объёме, часть обязательных элементов отсутствует или раскрыта недостаточно подробно. Работа отражает базовое понимание процедуры тестирования без системного подхода. 2 балла (неудовлетворительно). Тестирование выполнено фрагментарно. Тест-кейсы носят формальный характер или отсутствуют. Выявление ошибок нерегулярное, журнал дефектов не оформлен либо оформлен частично. Исправления выполняются без повторного тестирования. Использование Git ограничивается несколькими коммитами без логической структуры. Отчёт неполный, без анализа качества системы. Работа не демонстрирует	
--	--	--	--	--	--	--

			(Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — ISBN 978-5-507-52472-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/451820 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей. — С. 6.).
2	Основная литература	ЭБС издательства Лань	«Хортон, А. Разработка веб-приложений в ReactJS / А. Хортон, Р. Вайс ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2016. — 254 с. — ISBN 978-5-94074-819-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/97339 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей.» (Хортон, А. Разработка веб-приложений в ReactJS / А. Хортон, Р. Вайс ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2016. — ISBN 978-5-94074-819-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/97339 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей. — С. 1.).
3	Основная литература	ЭБС издательства Лань	«Разработка фронт-енд части веб-ориентированного информационного обеспечения систем управления в организационных системах : учебное пособие / В. Н. Волков, А. В. Демидов, С. В. Новиков [и др.]. — Орел : ОГУ имени И.С. Тургенева, 2023. — 427 с. — ISBN 978-5-9929-1541-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/451013 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей.» (Разработка фронт-енд части веб-ориентированного информационного обеспечения систем управления в организационных системах : учебное пособие / В. Н. Волков, А. В. Демидов, С. В. Новиков [и др.]. — Орел : ОГУ имени И.С. Тургенева, 2023. — ISBN 978-5-9929-1541-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/451013 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей. — С. 4.).
4	Дополнительная литература	ЭБС издательства Лань	Булгакова, И. А. Разработка и прототипирование веб-сайтов и интерфейсов онлайн: технологии создания интерфейсов : учебное пособие для вузов / И. А. Булгакова. — Санкт-Петербург : Лань, 2025. — 224 с. — ISBN 978-5-507-52858-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/502477 (дата обращения: 03.12.2025). — Режим доступа: для авториз. пользователей

Перечень используемого программного обеспечения:

1. PostgreSQL Team-PostgreSQL(бессрочно)
2. -XAMPP freeware(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Лекции	144 (36)	ПК, ПО, проектор, Интернет
Зачет	450 (36)	ПК, ПО, проектор, Интернет
Практические занятия и семинары	450 (36)	ПК, ПО, проектор, Интернет